Generic: parameterized structures .   
Vi du:   
- tao 1 list chua ds Student : List<Student> ds , Student la parameter type  
- tao 1 list chua cac chuoi ky tu : List<String> ds , String la parameter type  
- tao 1 list chua cac so nguyen :   
List<int> ds , int la parameter type (C#)  
List<Integer> ds , Integer la parameter type (Java)

Java: parameter type => reference type

C#: parameter type => value + reference type

**GENERICs**

1. ***Dinh nghia*** (definition) : co tinh tong quat – chung chung

Student<T>{  
T id;  
string name;

}

1. ***Khai thac / su dung***  (declare) : co tinh cu the , ie. tuong minh thay the tham so kieu T thanh 1 kieu nao do.

Student<string> sv1 = new Student(id=”s01”, name=”vuong”)  
Student<int> sv2 = new Student(id=120, name=”sang”)

Thi thuc hanh:

* Cac kien thuc co ban OOP: inheritance, abstract, interface
* Property (rw / r-only/ w-only ) + Regex
* Collection : ArrayList / List<> / Dictionary<,> / SortedList<, > / Hashtable
* Delegate / Event
* Iterator